AI OPTICS:OBJECT RECOGNITION AND CAPTION GENERATION FOR BLINDS USING DEEP LEARNING METHODOLOGIES

A Mini Project Report

submitted by

MOHAMMED AFNAN PP(MES20MCA-2026)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

February 2022

DECLARATION

I undersigned hereby declare that the project report **AI OPTICS:OBJECT RECOGNITION AND CAPTION GENERATION FOR BLINDS USING DEEP LEARNING METHODOLOGIES**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of Vasudevan.T.V, Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:Kuttippuram

.....

Date:02-03-2022

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **AI OPTICS:OBJECT RECOGNITION AND CAPTION GENERATION FOR BLINDS USING DEEP LEARNING METHODOLOGIES** is a bonafide record of the Mini Project work carried out by **MOHAMMED AFNAN PP(MES20MCA-2026)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

My endeavor stands incomplete without dedicating my gratitude to a few people who have contributed towards the successful completion of my project.We pay our gratitude to the Almighty for his invisible help and blessing for the fulfillment of this work.At the outset we express our heartful thanks to our Head of the Department and guide Prof. Vasudevan T.V for his valuable guidance and supervision. I take this opportunity to express our profound gratitude to Mr. Syed Feroze,group tutor and project coordinator Mrs.Priya JD for her valuable support, timely advise and strict schedules to complete project. I also grateful to all teaching and non-teaching staff for their encouragement, guidance and whole-hearted support. Last but not least, I gratefully indebted to family and friends, who gave me a precious help in doing our project

MOHAMMED AFNAN PP(MES20MCA-2026)

Abstract

At present many blind assistive systems have been implemented but there is no such kind of good system to navigate a blind person and also to track the movement of a blind person and rescue him/her if he/she is lost. In this paper, we have presented a blind assistive and tracking embedded system. In this system the blind person is navigated through a spectacle interfaced with an android application. The blind person is guided through English voice commands generated by the application according to the obstacle position. Using voice command a blind person can establish video call to a predefined number without touching the phone just by pressing the headset button. The blind assistive application gets the latitude and longitude using GPS and then sends them to a server. The movement of the blind person is tracked through another android application that points out the current position in Google map. We took distances from several surfaces like concrete and tiles floor in our experiment where the error rate is 5

Keywords: *AI OPTICS*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Blind mobility is one of the main brainstorming challenges that scientists are still facing around different parts of the world and still researching to implement suitable blind assistive devices. In recent years blind mobility has become an important issue since a large number of

people are visually impaired and partially sighted. Navigating a blind person is a great chal- lenge as blind person has to rely on other. The simplest and most widely used travelling aid

used by all blinds is the white cane. It has provided those people with a better way to reach destination and detect obstacles on ground, but it cannot give them a high guarantee to protect themselves from all level of obstacles. Sometimes it happens that blind people are lost and their guardians are in tension about them. There have been many efforts but even now, it is not easy for the blind people to move independently from one place to another. To solve this great problem we proposed a system, the system where a blind person can move without the help of other and can make emergency call to a predefined number and we can find out him/her easily if he/she is lost.

1.1 Background

This project is an android cum web application.

1.1.1 Motivation

The ANGEL Eye android mobile application is presented in this paper. The application provides assistance to visually impaired people by providing a set of useful features: Location tracking, Emergency help, object recognition. It has a user friendly interface customized for blind people. These various features can be provided through a single device, which reduces costs and complexity, and increases the practicality of the application. The presented results show that the proposed application successfully achieves its aims by providing the desired features.

1.2 Objective

Millions of people around the world face major disability of visual impairment. Vision provides all the information needed for reading, body movement, mobility and its loss can severely affect an individual's professional and social advancement. It was reported by the World Health Organization (WHO) that out of 1.3 billion people that suffer from one or another form of visual impairment, 36 million suffer from complete blindness. Problems are often faced by people with impaired vision or complete blindness once they are out of their familiarized environments. Corporeal development is one of the major issues for the people suffering from impaired vision. They also are unable to recognize an object without physically feeling it and can't savor the beauty of the nature. Many assistive devices have been made commercially available for the visually impaired community of the society to help them read and recognize objects, enhancing their experience. In this paper we propose and end-to-end accessible solution to provide purposeful acknowledgement and guidance using object recognition and caption generation for enabling video to audio aid for the visually impaired community of the society. The aim of purposeful acknowledgement and guidance is to extract the range and direction of the obstacles within a finite and defined free space captured by the camera of the device. The object detection and recognition algorithm will be fed results to the caption generation algorithm for explaining the scene of the surrounding to the visually impaired in audio format.

1.3 Report Organization

The project report is divided into four sections. Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

There are about 2.2 billion people with vision impairment or blindness. Visually impaired people face a lot of challenges in their day-to-day life. Physical movement is, in fact, one of the greatest challenges. Blind people find it very difficult to merely walk around high traffic areas or any unfamiliar places and therefore have to take help from other sighted individuals. To move around in familiar places, visually impaired people usually memorize the area and where the things are kept. However, those things if moved to some other place might cause issues for blind people. The most widely used way blind people use to travel from one place to other is the white cane. With all the technological developments, certain devices have been developed to help visually impaired people to move freely around in the environment. This paper presents a systematic literature review for some of the devices developed for navigation purpose of blind people. According to a survey conducted by WHO (World Health Organization) estimates that in the world, about 1human population is visually impaired and amongst them, about 10main problem with blind people is how to navigate their way. Many people with serious visual impairments cannot travel independently due to the heavy traffic on the road and in unfamiliar places. Blind people rely on lot of means on their mobility. Different approaches exist to help the visually impaired, the most widely used means for the mobility of the visual impaired people is the white cane with a red tip. It also enables to let people around them to know that the person is blind and help them to overcome obstacles. A long cane can also be used to extend the user's range of touch sensation. The cane is usually swung in a low sweeping motion, across the intended path of travel to detect the obstacles. Or they may use guide dogs which are characterized by a many imperfections. Even after so much

technological advancements there have not been many inventions for the help of visual impaired people. There are fewer types of equipment which fulfill their needs or provides them with the guidance. A.obstacle detection - Cameras can be used for detecting objects. Then once the object is detected then the visually impaired person must be alerted about it.when an obstacle is detected the blind person is let known through the voice message. With the rapid advances of modern technology both in hardware and software it has become easier to provide intelligent navigation system to the visually impaired.

Chapter 3

Methodology

3.1 Introduction

In this project I am developing a web and android application for assisting the blinds.Basic objective of this system is the blind person is navigated through a spectacle interfaced with an android application.The blind person is guided through Bengali/English voice commands generated by the application according to the obstacle position. This project contains main modules like Caretaker and Blind.Caretaker can register and they can login, they can add the blind under them through registration. It is the caretaker who can track the location of the blind through GPS tracking.Caretaker is the person who have the control over the blind by already pre-setting the app.Also the caretaker can get the emergency notification send by the blind. Blind can send voice command.Using voice command a blind person can establish voice call to a predefined number without touching the phone just by pressing the headset button. Also the blind updated their current location place to place that points out the current position in Google map. Blind can also detect the object.

Mainly there are two technical sides they were: Face recognition is a method of identifying or verifying the identity of an individual using their face. Recognition of familiar faces

involves a match between the products of structural encoding and previously stored structural codes describing the appearance of familiar faces, held in face recognition units. In my project facial recognition is used to help the blind users to recognise their familiar persons. I have used dlib,cmake libraries of python in-order to implement face recognition. Another important feature of the project is object recognition. In order to help the blind people to identify

various objects I have used Convolutions neural network. A Collection of training images was fed to the algorithm which finally resulted in a trained graph so that the testing was easier when uploaded a test image. A convolutions neural network is a class of artificial neural network, most commonly applied to analyze visual imagery, Which provides high accuracy test results.

3.2 Agile Methodology

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating. Continuous collaboration is vital, both with team members and project stakeholders. The project is divided into 4 sprints

Sprint1

– Data set collection from kaggle website

– preprocessing

Sprint2

– UI Designing

– Openpose library configuration

Sprint3

– Collecting user inputs

– coding

Sprint4

– Testing – Output Generation

Chapter 4

Results and Discussions

4.1 Datasets

The term data set refers to a file that contains one or more records. The record is the basic unit of information used by a program running on OS. Any named group of records is called a data set. Data sets can hold information such as medical records or insurance records, to be used by a program running on the system. Data sets are also used to store information needed by applications or the operating system itself, such as source programs, macro libraries, or system variables or parameters. For data sets that contain readable text, you can print them or display them on a console (many data sets contain load modules or other binary data that is not really printable). Data sets can be cataloged, which permits the data set to be referred to by name without specifying where it is stored. In simplest terms, a record is a fixed number of bytes containing data. Often, a record collects related information that is treated as a unit, such as one item in a database or personnel data about one member of a department. The term field refers to a specific portion of a record used for a particular category of data, such as an employee39;s name or department. The records in a data set can be organized in various ways, depending on how we plan to access the information. If you write an application program that processes things like personnel data, for example, your program can define a record format for each person39;s data. There are many different types of data sets in OS, and different methods for accessing them. Among the most commonly used types are: Sequential In a sequential data set, records are data items that are stored consecutively. To retrieve the tenth item in the data set, for example, the system must first pass the preceding nine items. Data items that

must all be used in sequence, like the alphabetical list of names in a classroom roster, are best stored in a sequential data set. Partitioned A partitioned data set or PDS consists of a directory and members. The directory holds the address of each member and thus makes it possible for programs or the operating system to access each member directly. Each member, however, consists of sequentially stored records. Partitioned data sets are often called libraries. Programs are stored as members of partitioned data sets. Generally, the operating system loads the members of a PDS into storage sequentially, but it can access members directly when selecting a program for execution. VSAM In a Virtual Storage Access Method (VSAM) key sequenced data set (KSDS), records are data items that are stored with control information (keys) so that the system can retrieve an item without searching all preceding items in the data set. VSAM KSDS data sets are ideal for data items that are used frequently and in an unpredictable order.

4.2 Results

The Output were generated

Chapter 5

Conclusions

In this paper, we have implemented a system for blind photo capturing and navigation. The final result of our approach is a blind navigation and tracking system with a flexible architecture that can be adopted in blind mobility. The system has been tested through several test cases. The blind assistive device with AngelEye for Blind android application is very useful for a blind person to move without the help of other and the user can seek emergency help through voice call. The tracking system involved here through BlindTracker application is very applicable to track the current location of the blind person.

References

[1] Johann B. and Iwan U. “The Guide Cane — A Computerized Travel Aid for the Active Guidance of Blind Pedestrians”, Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, NM, Apr. 21-27, 1997, pp. 1283-1288.

Appendix

Source Code

```
   import pymysql
from flask import *
import datetime
# from src.encode_faces import enf

from werkzeug.utils import secure_filename

app=Flask(__name__)
con=pymysql.connect(host='localhost',user='root',port=3307,password='root',db='angeleye')
cmd=con.cursor()
app.secret_key='ab'

@app.route('/')
def start():
   session.clear()
   return render_template('ADMIN.html')
@app.route('/adminhome')
def adminhome():
   return render_template('ADMINHOME.html')

@app.route('/login',methods=['post','get'])
def login():
   un=request.form['textfield']
   pwd=request.form['textfield2']
   cmd.execute("select * from login where username='"+un+"' and password='"+pwd+"'")
   s=cmd.fetchone()
   print(s)
   if s is None:
      return'''<script> alert("invalid");window.location="/"</script>'''
   # elif s[3]=="admin":
   #    session['lid'] = s[0]
   #    return '''<script> alert("admin login");window.location="/adminhome"</script>'''
   elif s[3] == "caretaker":
      session['lid'] = s[0]
      return '''<script> alert("caretaker login");window.location="/caretakerhome"</script>'''
   else:
      return '''<script> alert("invalid");window.location="/"</script>'''



# ________________caretaker_____________________________________________________________
```

Appendix

```
@app.route('/caretakerhome')
def caretakerhome():
 if 'lid' in session:
    return render_template('CARETAKERHOME.html')
 else:
    return '''<script> alert("PLEASE LOGIN");window.location="/"</script>'''


@app.route('/viewlocationofblind')
def viewlocationofblind():
 if 'lid' in session:
    cmd.execute("SELECT location.*,blind_person.`First_Name`,`blind_person`.`Last_Name` FROM `blind_person` JOIN`location`
          ON location.blind_id=blind_person.V_id")
    s=cmd.fetchall()
    return render_template('VIEWLOCATIONOFBLIND.html',val=s)
 else:
    return '''<script> alert("PLEASE LOGIN");window.location="/"</script>'''


@app.route('/blindpersonmanagement')
def blindpersonmanagement():
 if 'lid' in session:
    id=session['lid']
    cmd.execute("select * from blind_person where care_id='"+str(id)+"'")
    qry=cmd.fetchall()
    return render_template('BLINDPERSONMANAGEMENT.html',val=qry)
 else:
    return '''<script> alert("PLEASE LOGIN");window.location="/"</script>'''


@app.route('/editblind')
def editblind():
 if 'lid' in session:
    id=request.args.get('id')
    session['v_id']=id
    cmd.execute("select * from blind_person where v_id='"+str(id)+"'")
    qry=cmd.fetchone()
    return render_template('BLINDEDIT.html',val=qry)
 else:
    return '''<script> alert("PLEASE LOGIN");window.location="/"</script>'''

@app.route('/editperson',methods=['get','post'])
def editperson():
 if 'lid' in session:
    vid=session['v_id']
    FIRST_NAME=request.form['textfield']
    LAST_NAME=request.form['textfield2']
    GENDER=request.form['radiobutton']
    DOB=request.form['textfield3']
    PLACE=request.form['textfield8']
    POST=request.form['textfield4']
    PIN=request.form['textfield5']
    PHONE=request.form['textfield6']
    IMEI_NO=request.form['textfield7']
    phn=request.form['textfield18']

    cmd.execute("UPDATE `blind_person` SET
          `First_Name`='"+FIRST_NAME+"',`Last_Name`='"+LAST_NAME+"',`Gender`='"+GENDER+"',`DOB`='"+DOB+"',`Place`='"+PLACE+"',`Post`='"+POST+"',
          ,care_phnum='"+phn+"' WHERE `V_id`='"+str(vid)+"'")
```

Appendix

```
    print("UPDATE `blind_person` SET
          `First_Name`='"+FIRST_NAME+"',`Last_Name`='"+LAST_NAME+"',`Gender`='"+GENDER+"',`DOB`='"+DOB+"',`Place`='"+PLACE+"',`Post`='"+POST+"',
          ,care_phnum='"+phn+"' WHERE `V_id`='"+str(vid)+"'")
    con.commit()
    return'''<script> alert("success"); window.location="/caretakerhome" </script>'''
 else:
    return '''<script> alert("PLEASE LOGIN");window.location="/"</script>'''


@app.route('/deleteblind')
def deleteblind():
 if 'lid' in session:
    id=request.args.get('id')
    cmd.execute("delete from blind_person where V_id='"+str(id)+"'")
    con.commit()
    return'''<script> alert("deleted"); window.location="/caretakerhome" </script>'''
 else:
    return '''<script> alert("PLEASE LOGIN");window.location="/"</script>'''


@app.route('/blindregister', methods=['get','post'])
def blindregister():
 if 'lid' in session:
    return render_template('BLINDREGISTER.html')
 else:
    return '''<script> alert("PLEASE LOGIN");window.location="/"</script>'''

@app.route('/blindreg',methods=['get','post'])
def blindreg():
 if 'lid' in session:
    FIRST_NAME=request.form['textfield']
    LAST_NAME=request.form['textfield2']
    GENDER=request.form['radiobutton']
    DOB=request.form['textfield3']
    PLACE=request.form['textfield8']
    POST=request.form['textfield4']
    PIN=request.form['textfield5']
    PHONE=request.form['textfield6']
    IMEI_NO=request.form['textfield7']
    PHN=request.form['textfield18']

    lid=session['lid']
    cmd.execute("insert into blind_person
          values(Null,'"+FIRST_NAME+"','"+LAST_NAME+"','"+GENDER+"','"+DOB+"','"+PLACE+"','"+POST+"','"+PIN+"','"+PHONE+"','"+IMEI_NO+"','"+str(
    con.commit()
    return '''<script> alert("success"); window.location="/blindpersonmanagement" </script>'''
 else:
    return '''<script> alert("PLEASE LOGIN");window.location="/"</script>'''




@app.route('/viewemergencyhelp')
def viewemergencyhelp():
 if 'lid' in session:
    cmd.execute("SELECT `blind_person`.*,`help`.* FROM `help` JOIN `blind_person` ON `blind_person`.`imei_no`=`help`.imei")
    s=cmd.fetchall()
    return render_template('VIEWEMERGENCYHELP.html',val=s)
 else:
    return '''<script> alert("PLEASE LOGIN");window.location="/"</script>'''
```

Appendix

```
if __name__ ==’__main__’:
   app.run(debug=True)

   # web service.py
   import pymysql
import time
from flask import *
from scipy.ndimage import rotate
from src.database import select, iud, selectall


from src.recognize_face import rec_face_image
import cv2




con = pymysql.connect(host="localhost", port=3307, user="root", password="root", db=’angeleye’)
cmd = con.cursor()

face_cascade = cv2.CascadeClassifier(’haarcascade_frontalface_default.xml’)
app = Flask(__name__)
from src.objectdetection import objdet

@app.route(’/updateloc’, methods=[’POST’])
def updateloc():
   try:
      print(request.form)
      imei = request.form[’imei’]
      latt = request.form[’latt’]
      long = request.form[’long’]

      print(long)
      print(imei)
      qry = ’SELECT `V_id` FROM `blind_person` WHERE `IMEI_NO`=%s’
      res = select(qry, (str(imei),))
      print(res)
      qry="select * from location WHERE `blind_id`=%s"
      val = (str(res[0]))

      ss=select(qry,val)
      print("ss",ss)
      if ss is None:
         print("okkkkkkkkkkkkkkkkkkkkkk")
         qry = ’insert into location values(null,%s,%s,%s)’
         val = ( str(res[0]),str(latt), str(long))
         iud(qry, val)
         return jsonify({"task":"ok"})
      else:
         # long="433443"


         print("alrdyyyyyyyyyyy")
         query="select * from location WHERE blind_id=%s and latitude=%s and longitude=%s "
         val = (str(res[0]), str(latt), str(long))
         sss=select(query,val)
         print(val)
         if sss is None:
            print("latiiiiiiiilongii")
```

Appendix

```
            qry = 'UPDATE `location` SET `latitude`=%s, `longitude`=%s WHERE `blind_id`=%s'
            val = (str(latt), str(long), str(res[0]))
            iud(qry, val)

            result= select(query, val)
            print ("result",result)

            return jsonify({"task": "ok"})
        return jsonify({"task": "ok"})


    except Exception as e:
        print(e)
        return jsonify({"task":"error"})
@app.route('/photoupload', methods=['POST','GET'])
def photoupload():
    try:
        print(request.files)


        file=request.files['file']
        print(file)
        timestr = time.strftime("%Y%m%d-%H%M%S")

        try:
            filename = timestr + "frame.jpg"
            cv2.imwrite(filename)


            # cv2.imwrite(timestr+"frame.jpg", frame) # save frame as JPEG file
        except Exception as e:
            print(e)

        return jsonify({'task':';;;;;;;;;;'})

    except:

        return jsonify({'task':"No result"})
#

@app.route('/addemergency', methods=["POST"])
def addemergency():
    try:
        imei = request.form['IMEI_NO']
        latt = request.form['latitude']
        long = request.form['longitude']
        qry = 'SELECT `V_id` FROM `blind_person` WHERE `imei`=%s'
        res = select(qry, (str(imei),))
        qry = 'INSERT INTO `help` VALUES(NULL, %s, CURDATE(), %s, %s)'
        val = (str(res[0]), str(latt), str(long))
        iud(qry, val)
        return jsonify({'task': 'ok'})
    except Exception as e:
        print(e)
        return jsonify({'task': 'error'})


@app.route('/capture', methods=["POST"])
def capture():
    try:
        print(request.files)
```

Appendix

```
save_location = "static/identify.jpg"
img = request.files['files']
imei=request.form['imei']
print(imei)
img.save(save_location)

img=cv2.imread(save_location)
print(type(img))
try:
   img=rotate(img, 270)
   cv2.imwrite("ro.jpg",img)
except Exception as e:
   print(e)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# faces = face_cascade.detectMultiScale(gray, 1.3, 5)
#
# print("pppppppppppp",faces)
# print("lngthhhhhhhhhhh",len(faces))
#


# if len(faces)==0:
results = objdet(save_location)
re=', '.join(results)+" are in front of you"
print(re)

return jsonify({'task': re})
# else:
#    print("pppppppppppppppppppppppppp")
#    cmd.execute("SELECT * FROM `familiar_person`")
#    s = cmd.fetchall()
#    from src.encode_faces import enf
#    enf("ro.jpg")
#
#    for r in s:
#
#       res = rec_face_image("static/familiar_prsn/" + r[3])
#       print("R", res)
#       if res is not None:
#          print("GETTTTTTTTTTTTTTTTTTTTTTTTTTTT", res)
#          cmd.execute("select * FROM `familiar_person` WHERE `image`='" + str(res) + "'")
#          s = cmd.fetchone()
#          print(s)
#          print({'task': " " + s[2] + " is infront of you " + "relation is " + s[4]})
#    return jsonify({'task':" "+s[2]+" is infront of you"+"relation is"+s[4]})



         # print("okkkkkkkkkkkkbjfdzsfxghjkl;")
   # ids=rec_face_image(save_location)
   # print("ooooooooooooooo",ids)
   #
   #
   # qry="SELECT NAME, relation FROM `familiar_person` WHERE `blind_id` IN(SELECT `V_id` FROM `blind_person` WHERE
        `IMEI_NO`='75f6f1d87cf206f3') AND `familiar_person`.`f_id` =4"
   # print(qry)
   #
   # s=selectall(qry)
   # if len(s)==0:
   #    return jsonify({'task': " person is infront of you"})
   # print(s)
   # return jsonify({'task':" "+s[0][0]+" is infront of you"+"relation is"+s[0][1]})
```

```
    except Exception as e:
        print(e,"====================================")
        return jsonify({'task': "unknown person recognized"})
        return jsonify({'task': "unknown person recognized"})

#
# @app.route('/ocr', methods=["POST"])
# def ocr():
#     try:
#         file_location = './static/ocr.jpeg'
#         img = request.files['files']
#         img.save(file_location)
#         result_text=main(file_location)
#         return jsonify({'task': result_text})
#     except Exception as e:
#         print(e)
#         return jsonify({'task': 'error'})

@app.route('/getphone', methods=["POST"])
def getPhone():
    try:
        print(request.form)
        imei = request.form['IMEI_NO']
        imei="df58368c83ea3b84"
        qry = 'SELECT `caretaker`.`phone`, `blind_person`.`First_Name`, `blind_person`.`Last_Name FROM `blind_person` JOIN
              `caretaker` ON `blind_person`.`care_id` = `caretaker`.`Login_id` WHERE `blind_person`.`imei`= %s'
        res = select(qry, (str(imei),))
        return jsonify({'status': 'ok', 'phone': str(res[0]), 'bname':res[1]})
    except Exception as e:
        print(e)
        return jsonify({'status': 'error'})

# Route to recieve volunteer phone number with highest number of points
@app.route('/getvolphone', methods=['GET','post'])
def getvolphone():
    try:
        # print(request.form,"====================")
        imei = request.form['IMEI_NO']
        lat = request.form['latti']
        longi = request.form['logi']
        print(imei,"imeiiiiiiiiiii")

        print(imei)
        qry ="SELECT `care_phnum` FROM `blind_person` WHERE IMEI_NO= %s"
        res = select(qry, (str(imei),))
        print(";lkjhgdfsafgjhkjlk;",res)

        volPhone = str(res[0])
        qrys = "INSERT INTO `help` VALUES(NULL,%s,CURDATE(),%s,%s)"
        vall = (imei, str(lat), str(longi))
        iud(qrys, vall)

        print(res,"======>")
        return jsonify({
            'volphone': volPhone
        })
    except:
        print(request.form, "====================")
        imei = 'e91b44dcc8cac7f3'
        qry ="SELECT `care_phnum` FROM `blind_person` WHERE IMEI_NO= %s"
        res = select(qry, (str(imei),))

        lat = request.form['lati']
```

Appendix

```
        longi = request.form['longi']

        volPhone = str(res[0])

        # qrys = "INSERT INTO `help` VALUES(NULL,%s,CURDATE(),%s,%s)"
        # vall = (imei, str(lat), str(longi))
        # iud(qrys, vall)
        print(res, "======>")
        return jsonify({
            'volphone': volPhone
        })



app.run(host='0.0.0.0',port=5000)
def sample():
    print("pppppppppppppppppppppppp")
    cmd.execute("SELECT * FROM `familiar_person`")
    s = cmd.fetchall()
    from src.encode_faces import enf
    enf("ro.jpg")

    for r in s:

        res = rec_face_image("static/familiar_prsn/" + r[3])
        print("R", res)
        if res is not None:
            print("GETTTTTTTTTTTTTTTTTTTTTTTTTTT", res)
            cmd.execute("select * FROM `familiar_person` WHERE `image`='" + str(res) + "'")
            s = cmd.fetchone()
            print(s)
            print({'task': " " + s[2] + " is infront of you " + "relation is " + s[4]})
    print("LOOOOOOOOOOOOOOOST")

# sample()
```

Database Design

Attribute Name	Datatype	Width	Description
id	Integer	10	Primary Key
Username	Varchar	20	
Password	Varchar	10	

Table A.1: Login

Attribute Name	Datatype	Width	Description
c id	Integer	10	Primary key
Login id	Varchar	20	
First Name	Varchar	20	

Table A.2: Caretaker

Attribute Name	Datatype	Width	Description
help id	Integer	10	Primar key
Imei	Varchar	20	
Date	DateNotnull	10	

Table A.3: Help

DaTaflow Diagram

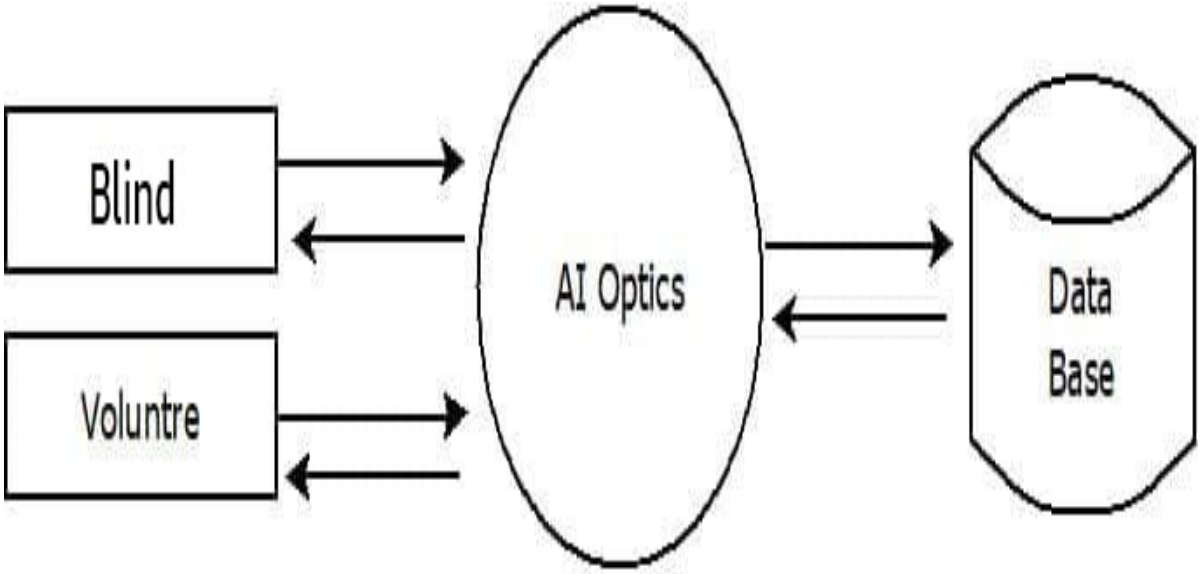

Figure A.1: Level0

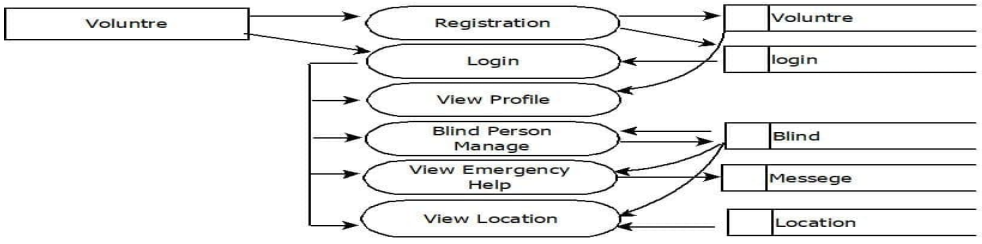

Figure A.2: Level 1.1

Figure A.3: Level1.2

Screen Shots

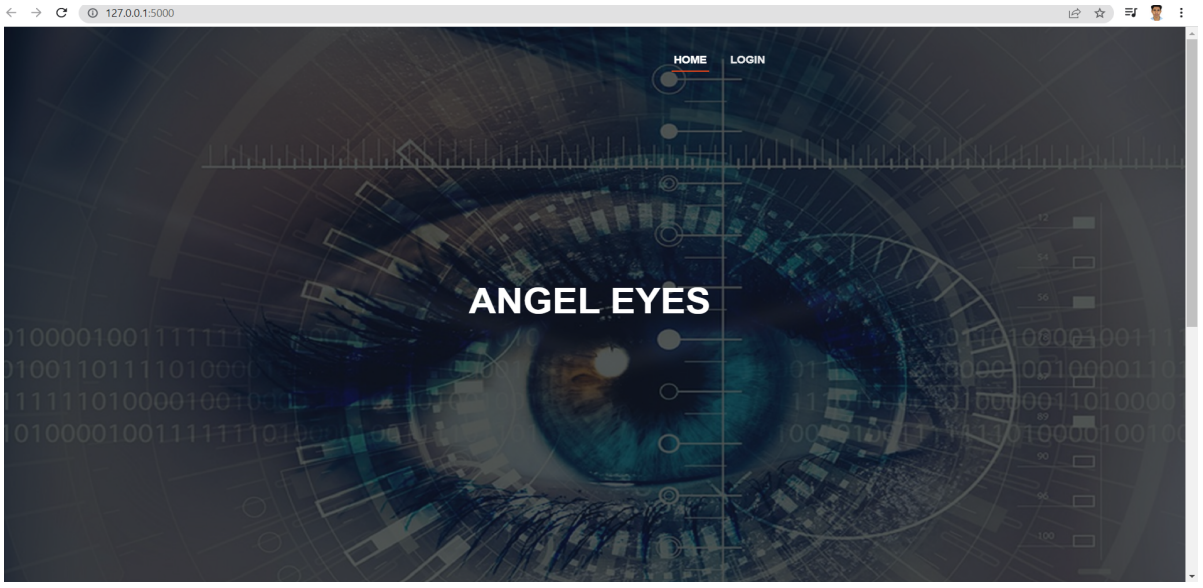

Figure A.4: User interface1

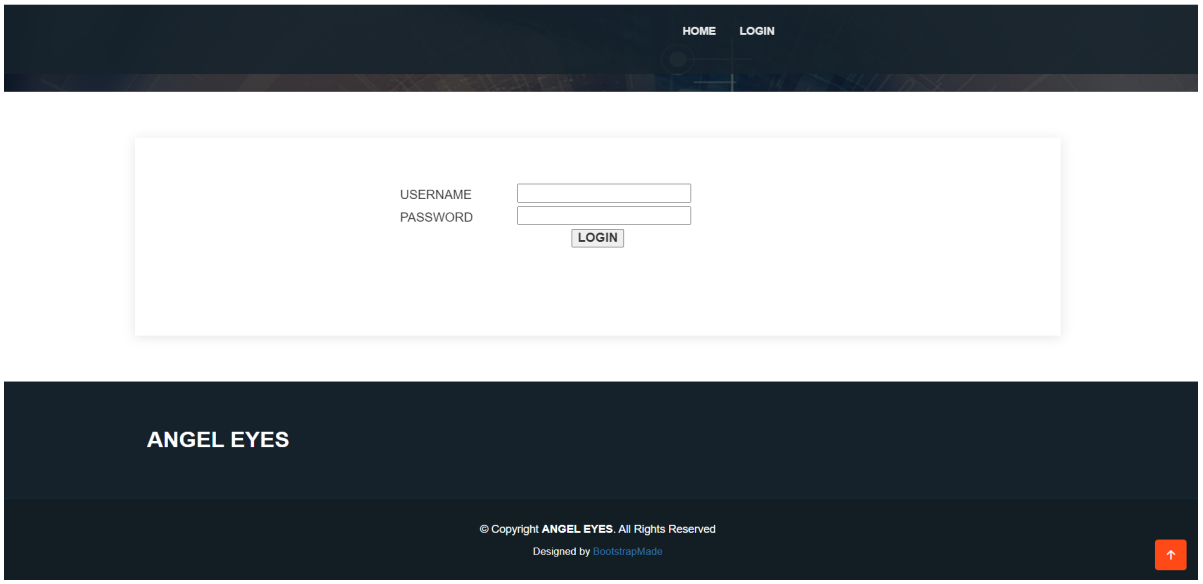

Figure A.5: User Interface2

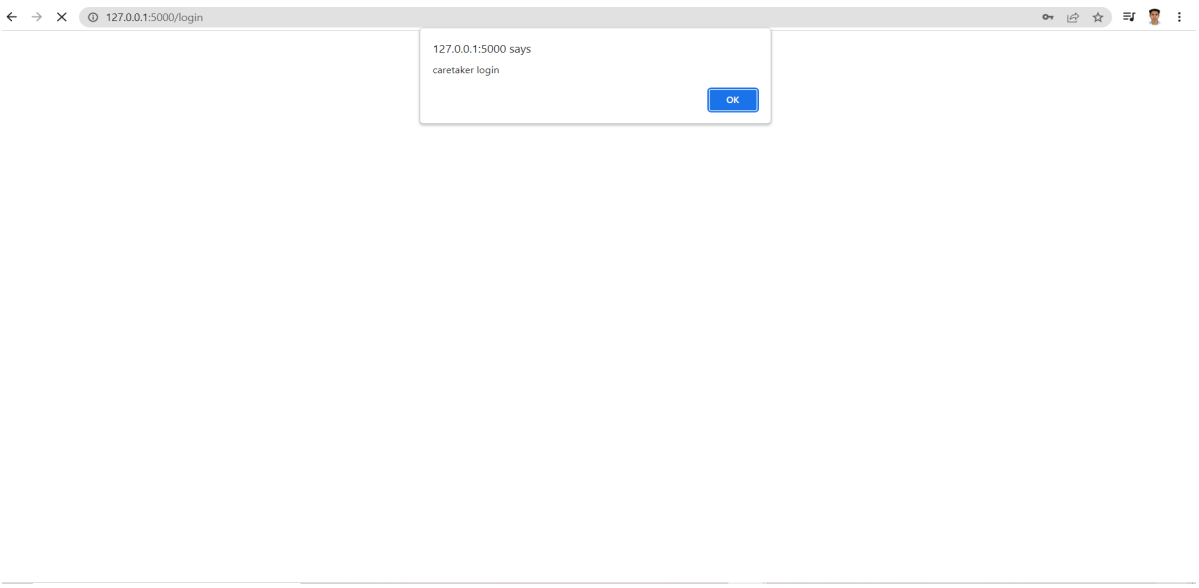

Figure A.6: User Interface3

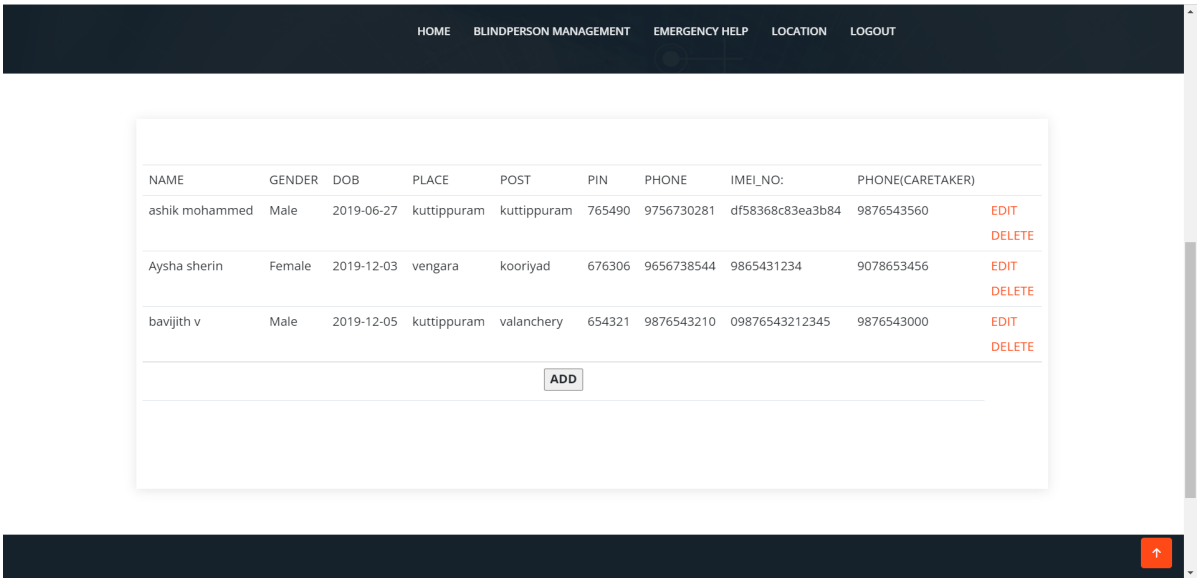

Figure A.7: User Interface4

User Story

USER STORIES

UserStoryID	As a <type of user>	I want to	So that I can
1	Table design	Design tables for project	Create tables with normalization
2	Form design	Complete form design	Complete form designs for our project
3	Linking	Load and link html pages	Complete load and link
4	Volunteer	login	login successful with correct username and password
5	Volunteer	Update Profile	View and update all the personal details.
6	Volunteer	View emergency help	view blind persons emergency help and track the current location.

Figure A.8: User story

Project Plan

PROJECT PLAN

User story ID	Task name	Start date	End date	Days	Status
Sprint1					
1	UI designing	27/11/2021	30/11/2021	3	Completed
2	Database connectivity	18/12/2021	20/12/2021	2	Completed
3	Coding	04/01/2022	08/01/2022	5	Completed
Sprint2					
4	UI designing	13/01/2022	16/01/2022	3	Completed
5	Database connectivity	10/01/2022	14/01/2022	4	Completed
6	Coding	27/01/2022	31/01/2022	5	Completed

Figure A.9: Project plan 1

User story ID	Task name	Start date	End date	Days	Status
Sprint3					
7	UI designing	02/02/2022	05/02/2022	3	Completed
8	Database connectivity	06/02/2022	08/02/2022	2	Completed
9	Coding	10/02/2022	15/02/2022	5	Completed
10	Testing & validation	16/02/2022	17/02/2022	1	Completed
Sprint4					
11	UI designing	18/02/2022	20/02/2022	2	Completed
12	Database connectivity	20/02/2022	21/02/2022	1	Completed
13	Coding	20/02/2022	23/02/2022	5	Completed

Figure A.10: Project plan 2

Product Backlog and Sprint Backlog

PRODUCT BACKLOG

User Story ID	Priority <High/Medium/Low>	Size (Hours)	Sprint <#>	Status <Planned/In progress/Completed>	Release Date	Release Goal
1	Medium	2	1	Completed		Table design
2	High	3		Completed		Form design
3	High	5		Completed		Basic coding
3	High	5	2	Completed		Creation data set
4	Medium	5		Completed		Preprocessing
5	High	5	3	Completed		Training
6	medium	5		Completed		Voice Alert
7	Medium	5	4	Completed		Testing data
8	High	5		Completed		Output generation

Figure A.11: Product Backlog

SPRINT BACKLOG PLAN

Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story #1,#2,#3,#4		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Volunteer registration form	20/01/2022	5	1	1	0	0	0	0	0	1	0	0	1	0	0	1
Volunteer Login form	28/01/2022	5	0	0	0	1	1	1	0	0	0	0	0	1	0	1
Home page	03/02/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Volunteer	05/02/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Database operations	08/02/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Database operations for caretaker booking	12/02/2022	5	0	0	0	0	0	0	0	1	0	1	1	1	0	1
Testing	14/02/2022	5	0	1	0	0	1	0	0	1	0	0	1	0	0	1
Output generation	20/02/2022	5	0	0	0	0	0	0	0	0	0	2	1	1	1	1
Total		40	4	4	2	4	3	2	0	2	0	5	4	4	3	4

Figure A.12: Sprint Backlog

SPRINT ACTUAL

Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story #1,#2,#3,#4		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Volunteer registration	20/01/2022	5	1	1	0	0	0	0	0	0	1	0	0	1	0	1
Volunteer Login form	28/01/2022	2	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Home page	03/02/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Basic coding	08/02/2022	5	1	0	1	0	1	0	1	0	0	0	0	0	0	0
Caretaker Dataset	11/02/2022	3	1	0	0	0	1	0	0	0	0	1	0	0	0	0
Data Base operation	13/02/2022	5	1	0	0	1	0	0	0	1	0	0	1	0	0	1
Training	15/02/2022	5	1	0	1	1	0	0	0	1	0	0	0	0	1	0
Testing	18/02//2022	5	0	1	0	0	1	0	1	0	0	1	0	1	0	0
Output generation	20/02/2022	5	0	0	1	1	0	0	0	1	0	1	0	0	1	0
Total		40	1	1	0	1	1	1	0	0	0	1	1	1	1	1

Figure A.13: Sprint Actual